

APPLICATION FOR A UNITED STATES PATENT

UNITED STATES PATENT AND TRADEMARK OFFICE

(MBHB CASE No. 99,837-A; 3Com Case No. 2674.CS.US.P)

5 Title: **METHOD AND SYSTEM FOR INTEGRATING IP ADDRESS
RESERVATIONS WITH POLICY PROVISIONING**

10 Inventors: John G. Fijolek, a citizen of the United States and a resident of Naperville,
Illinois;
Irene Gilbert, a citizen of the United States and a resident of Palatine, Illinois;
15 Jaideep Kulkarni, a citizen of India, and a resident of , Illinois;
Steven Brant Levy, a citizen of the United States and a resident of Seattle,
Washington;

20 Attorney: George I. Lee
McDonnell Boehnen Hulbert & Berghoff
300 S. Wacker, 32 Floor
Chicago, IL 60606

25 Assignee: 3Com Corporation
5400 Bayfront Plaza
Santa Clara, CA 95052

REFERENCE TO RELATED APPLICATIONS

This application claims priority benefits to the United States provisional application Serial No. 60/240294 filed on October 13, 2000.

FIELD OF THE INVENTION

5 The present invention relates to communications in computer networks. More particularly, it relates to a method and a system for policy management and multiple access provisioning.

BACKGROUND OF THE INVENTION

10 Cable television networks such as those provided by Comcast Cable Communications, Inc., of Philadelphia, Pennsylvania, Cox Communications of Atlanta Georgia, Time-Warner Cable, of Marietta Georgia, Continental Cablevision, Inc., of Boston Massachusetts, and others, provide cable television services to a large number of subscribers over a large geographical area. The cable television networks typically are interconnected by cables such as coaxial cables or a Hybrid Fiber/Coaxial ("HFC") cable system which have data rates of about 10 Mega-bits-per-second ("Mbps") to 30+ Mbps.

15 The Internet, a world-wide-network of interconnected computers, provides multi-media content including audio, video, graphics and text that requires a large bandwidth for downloading and viewing. Most Internet Service Providers ("ISPs") allow customers to connect to the Internet via a serial telephone line from a Public Switched Telephone Network ("PSTN") at data rates including 14,400 bps, 28,800 bps, 33,600 bps, 56,000 bps and others that are much slower than the about 10 Mbps to 30+ Mbps available on a coaxial cable or HFC cable system on a cable television network.

20 With the explosive growth of the Internet, many customers have desired to use the larger bandwidth of a cable television network to connect to the Internet and other computer networks.

Cable modems, such as those provided by 3Com Corporation of Santa Clara, California, Motorola Corporation of Arlington Heights, Illinois, Cisco Corporation of San Jose, California, Scientific-Atlanta, of Norcross, Georgia and others offer customers higher-speed connectivity to the Internet, an intranet, Local Area Networks ("LANs") and other computer networks via cable television networks. These cable modems currently support a data connection to the Internet and other computer networks via a cable television network with a data rate of up to 30+ Mbps, which is a much larger data rate than can be supported by a modem used over a serial telephone line.

Many cable television networks provide bi-directional cable systems, in which data is sent "downstream", from a "headend" to a customer, as well as "upstream", from the customer back to the headend. The cable system headend is a central location in the cable television network and, further, is responsible for sending cable signals in the downstream direction and receiving cable signals in the upstream direction. An exemplary data-over-cable system with RF return typically includes customer premises equipment such a customer computer, a cable modem, a cable modem termination system, a cable television network, and a data network such as the Internet.

Some cable television networks provide only uni-directional cable systems, supporting only a "downstream" data path, which provides a path for flow of data from a cable system headend to a customer. A return data path via a telephone network, such as a public switched telephone network provided by AT&T and others, (i.e., a "telephone return") is typically used for an "upstream" data path, which provides a path for flow of data from the customer back to the cable system headend. A cable television system with an upstream connection to a telephone network is typically called a "data-over-cable system with telephone return."

An exemplary data-over-cable system with a telephone return typically includes customer premise equipment ("CPE") entities (such as a customer computer or a Voice over Internet Protocol ("VoIP") device), a cable modem, a cable modem termination system, a cable television network, a public switched telephone network, a telephone remote access concentrator, and a data network (e.g., the Internet). The cable modem termination system and the telephone remote access concentrator combined are called a telephone return termination system.

If the customer premises equipment entity comprises a telephone or a device capable of sending and receiving video or voice signals, the cable modem has to be capable of sending and receiving such signals. In such cases the cable modem typically comprises an internal media terminal adapter, which provides a network interface functionality that accepts analog voice inputs or video signal and generates IP packets using the Real Time Transport protocol, for instance.

In a bi-directional cable system, when the cable modem termination system receives data packets from the data network, the cable modem termination system transmits received data packets downstream via the cable television network to a cable modem attached to the customer premises equipment entity. The customer premises equipment entity sends response data packets to the cable modem, which sends the response data packets upstream via the cable network. The cable modem termination system sends the response data packets back to the appropriate host on the data network.

In the case of a telephone return system, when the cable modem termination system receives data packets from the data network, the cable modem termination system transmits the received data packets downstream via the cable television network to a cable modem attached to the customer premises equipment entity. The customer premises equipment entity sends response data packets to the cable modem, which sends response data packets upstream via the

public switched telephone network to the telephone remote access concentrator. Next, the telephone remote access concentrator sends the response data packets back to the appropriate host on the data network.

When a cable modem used in the cable system with the telephone return is initialized, a connection is made to both the cable modem termination system via the cable network and to the telephone return termination system via the public switched telephone network. As the cable modem is initialized, the cable modem initializes one or more downstream channels via the cable network. Also upon initialization, the cable modem receives a configuration file (a boot file) from a configuration server via a trivial file-transfer protocol ("TFTP") exchange.

The configuration file may include a plurality of configuration parameters encoded in a type-length-value format ("TLV"), for instance. The configuration file may comprise a plurality of Class-of-Service ("CoS") and Quality-of-Service ("QoS") parameters. The Class of Service parameters include, for example, maximum allowed rates, minimum reserved rate, maximum latency and a plurality of other parameters. The Quality of Service parameters include, for example, parameters defining delays expected to deliver data to a specific destination, the level of protection from unauthorized monitoring or modification of data, expected residual error probability, relative priority associated with data and a plurality of other parameters.

Upon receipt of the configuration file, a cable modem may register with a cable modem termination system. To do that, the cable modem may send to the cable modem termination system a registration request message comprising a copy of the configuration file including a plurality of QoS and CoS parameters.

Typically, thousands of cable modems are connected to each cable modem termination system, and also a plurality of customer premises equipment ("CPE") entities such as computers, VoIP compliant devices or telephones are connected to each cable modem. However, there are

several problems associated with providing access to subscription services for tens of thousands of cable modems and customer premises equipment entities. First, no generic methods exist for identifying each cable modem and customer premises equipment. Further, servers may not distinguish different classes of customer premises equipment connected to a cable modem and, thus, the servers may not dynamically provide different classes of service when a customer premises device associated with the cable modem requests, for example, an IP address.

Further, with a growing number of companies manufacturing cable modems, each cable modem associated with a specific vendor requires a unique boot file. Therefore, for example, if there were five cable modems produced by five different vendors, a DHCP server would have to store five unique boot files and, further, the server would have to be programmed with software to determine a type of device that had requested a boot file. However, as known in the art, DHCP servers are unable to distinguish the identity of cable modems or CPEs and, typically, each server stores only one boot file and sends the same boot file to all devices that request a configuration file.

Thus, it is desirable to provide a standard, reliable and efficient way to provide subscriber provisioning tools preferably integrated into the existing cable modem infrastructure. Further, it is desirable to develop a method and system for cable modem boot file management and IP service classes management.

SUMMARY OF THE INVENTION

According to an exemplary embodiment, a method and a system for policy management and multiple access provisioning are developed.

In one embodiment, a data-over-cable system for policy provisioning and access managing is developed. The exemplary system includes a first network device for marking an incoming message with an identifier of a network access device, a second network device for policy provisioning and access managing and a database for storing a plurality of configuration information records. According to an exemplary embodiment, the second network device intercepts the incoming message prior to at least one first protocol server such as a DHCP server receives the incoming message. Further, the second network device identifies a network device associated with the incoming message using the identifier in the first message and manages an assignment of configuration settings based on the identifier. In the exemplary embodiment, the database stores a plurality of configuration information records, and each record includes an identifier of a network access device and a plurality of configuration information settings constructed based on a service level agreement associated with the identifier of each record. In the exemplary embodiment, the identifier comprises a Medium Access Control address of the network access device. The configuration information settings include a service provider identifier with a path to a service provider associated with the identifier, a configuration file identifier with a path of a configuration file on a second protocol server such as a TFTP server and a class of service identifier. In the exemplary embodiment, the second network device uses the class of service identifier to redirect the incoming message and to assign an IP address to a network device in communication with the network access device from an IP address pool associated with the class of service parameter.

In one embodiment, a method for policy provisioning and access managing is developed. The method includes receiving a first message on a first network device from a second network device and marking the first message with an identifier of a network access device. In one embodiment, the identifier includes a MAC address of the network access device. Further, the method includes intercepting the first message on a third network device prior to at least one first protocol server receiving the first message. In one embodiment, the first protocol server comprises a DHCP server. The third network device determines the identity of the second network device using the identifier in the first message and manages the assignment of configuration parameters for the second network device based on the identifier of the network access device. In the exemplary embodiment, managing the assignment of the configuration parameters for the second network device includes establishing a plurality of configuration records in a database, and querying the database by the third network device for a record associated with the identifier in the first message. In one embodiment, each record includes an identifier of a network access device and a plurality of configuration settings associated with the identifier in the record.

According to an exemplary embodiment, the configuration settings include a configuration file identifier with a path of a configuration file on a second protocol server, the second protocol server including a TFTP server. In the exemplary embodiment, the third network device inserts the path of the configuration file in the record to the first message. Further, the configuration settings include an identifier of a first protocol service provider such as a DHCP service provider. In one embodiment, the third network device forwards the first message to a service provider specified in the record. Further, according to an exemplary embodiment, the configuration settings include class of service settings. In one embodiment, the third network device uses the class of service settings to provide a network device communicates



5

[illegible]

BRIEF DESCRIPTION OF THE DRAWINGS

Exemplary embodiments of the present invention are described with reference to the following drawings, in which:

Figure 1 is a block diagram illustrating a cable modem system in which exemplary
5 embodiments of the present invention may be applied;

Figure 2 is a block diagram illustrating a protocol stack for a data-over-cable system;

Figure 3 is a block diagram illustrating a Termination System Information message
structure;

Figure 4 is a block diagram illustrating a Dynamic Host Configuration Protocol message
10 structure;

Figures 5 is a block diagram illustrating an exemplary data-over-cable provisioning and
access managing system;

Figure 6 is an exemplary dialog box that a system administrator uses to administer and
manage a provisioning-access manager.

15 Figure 7 is an exemplary dialog box that a system administrator uses to manage
configuration files.

Figure 8 is an exemplary dialog box that a system administrator uses to manage class of
service settings;

Figure 9 is an exemplary dialog box that a system administrator uses to manage a
20 filtering mechanism of network devices on a data-over-cable system;

Figure 10 is an exemplary dialog box that a system administrator uses to manage
forwarding of messages on a data-over-cable network;

Figure 11 is an exemplary dialog box that a system administrator uses to manage and set
up service wide options;

Figure 12 is a flow chart illustrating an exemplary method for provisioning and access managing of a network device;

Figure 13A and 13B are a flow chart illustrating an exemplary method for provisioning and access managing of a cable modem;

5 Figure 14A and 14B are a flow chart illustrating an exemplary method for provisioning and access managing of customer premises equipment;

Figure 15 is a block diagram of a data-over-cable system for an automatic web registration; and

10 Figure 16A and 16B are a flow chart illustrating an exemplary method for automated provisioning and configuring a network device.

DETAILED DESCRIPTION
OF AN EXEMPLARY EMBODIMENT

Exemplary data-over-cable system

5 Figure 1 is a block diagram illustrating an exemplary data-over-cable system 10. The data-over-cable system 10 may be a bi-directional cable system supporting a downstream data flow and an upstream data flow to and from a cable television network "headend" from and to a customer premises equipment entity such as a personal computer, for instance. The cable television network "headend" is a central location responsible for sending cable signals in a downstream and an upstream direction. In a bi-directional cable system, customer premises equipment entities or a cable modem may have an upstream connection to a cable modem termination system via a cable television connection, a wireless connection, a satellite connection or a different connection by which the cable modem may send data upstream to the cable modem termination system.

10
15 Alternatively, the data-over-cable system 10 may be a uni-directional cable system supporting only a downstream data path from a cable television network headend to a customer premises equipment entity, such as a personal computer. In the uni-directional cable system, a return path is typically established via a telephone network ("telephone return"), which provides an "upstream" data path from the customer premises equipment back to the cable television network "headend". In a uni-directional cable system, a cable modem may comprise an integral telephone modem for connecting to a Public Switched Telephone Network ("PSTN") such as a PSTN 22, and the integral telephone modem may be connected to the cable modem for exchanging data.

20 The data-over-cable system 10 includes a Cable Modem Termination System ("CMTS") 12 connected to a cable television network 14, hereinafter a cable network 14. Figure 1 illustrates one CMTS 12. However, the data-over-cable system 10 may include multiple CMTS

12. Further, according to an exemplary embodiment, the CMTS 12 and any other network entities that will be described in the following paragraphs may be duplicated in a serial or a parallel arrangement to provide a back-up in case of failure.

In the exemplary embodiment of the present invention, the CMTS 12 may be a Total Control hub by 3Com Corporation of Santa Clara, California, with a cable modem termination unit. A Total Control hub is a chassis with multiple networking cards connected by a common bus. However, the CMTS 12 could also be another network server such as a network server by Cisco Systems of San Jose, California, for instance.

The cable network 14 may be a cable television network such as one provided by Comcast Cable Communications, Inc., of Philadelphia, Pennsylvania, Cox Communications of Atlanta, Georgia, or Time-Warner Cable, of Marietta, Georgia, for instance.

A cable modem ("CM") 16 is connected to the cable network 14 with a downstream cable connection. The cable modem may be provided by 3Com Corporation of Santa Clara, California, or Motorola Corporation of Arlington Heights, Illinois, for instance.

In the exemplary embodiment, the data-over-cable system 10 operates according to a set of specifications, one of which is the Data Over Cable Service Interface Specification ("DOCSIS"), published by Cable Television Laboratories. The DOCSIS standards may be found on the World Wide Web at the Universal Resource Locator ("URL") "www.cablemodem.com." As known in the art, the DOCSIS specification defines interface requirements for cable modems involved in a high speed data distribution over cable television networks. Further, the data-over-cable system 10 may be Packet Cable specifications compliant. The Packet Cable standards may be found on the World Wide Web at the URL "www.packetcable.com." The Packet Cable specifications define mechanisms required for supporting voice and video transmission over cable systems. If the data-over-cable system 10 is Packet Cable specification compliant, the CM

16 may comprise an internal media terminal adapter, or a media terminal adapter may otherwise be provided in communications with the CM 16. The media terminal adapter may provide a network interface functionality for transmitting voice or video signals and for converting analog voice inputs or video signals to IP packets using, for instance, the Real Time Transport protocol.

5 Furthermore, if the data-over-cable system 10 is Packet Cable Specification compliant, the data-over-cable system 10 may include a plurality of additional network devices such as a call management server and a gate controller, for instance. The call management server may enable the media terminal adapter to establish multimedia sessions including voice communications applications such as "IP telephony" or "VoIP". The gate controller may be used to perform authorization and authentication checks for users attempting to connect to the CMTS 12.

Figure 1 illustrates one CM 16 connected to the CMTS 12. However, typical data-over-cable systems include tens or hundreds of thousands of CMs 16 connected to the CMTS 12. In addition, as shown in Figure 1, the CM 16 is connected to a CPE entity 18 such as a personal computer system, a VoIP device or a telephone, for instance. The CM 16 may be connected to the CPE entity 18 via a Cable Modem-to-CPE Interface ("CMCI") 20. Figure 1 illustrates one CPE entity 18. However the CM 16 is typically coupled to multiple CPE entities.

If the data-over-cable system 10 is a bi-directional data-over-cable system, the CM 16 may have an upstream and downstream connection to the CMTS 12 via a cable television connection, a wireless connection or a satellite connection, for instance. Figure 1 illustrates an exemplary upstream and downstream connection to the CMTS 12 via the cable network 14. In such embodiment, the CMTS 12 may still also provide data from the PSTN 22 to the CM 16 or the CPE entity 18.

In a downstream direction of a bi-directional data-over-cable system, a cable system typically has a passband with a lower edge between 50 MHz and 54 MHz and an upper edge between 300 MHz to 864 MHz. However, the data-over-cable system 10 is not limited to such frequencies, and frequencies in data-over-cable system may be implementation dependent. In the upstream direction, the cable system may have an operating frequency passband range from 5 MHz to 30 MHz or 5 MHz to 40 MHz, for instance.

As mentioned above, the cable system 10 may be a unidirectional cable system. In a unidirectional cable system, the CM 16 is connected to the PSTN 22 or other such network, which provides an upstream telephone connection. The upstream telephone connection may be a standard telephone line connection such as an Integrated Services Digital Network ("ISDN") connection, an Asymmetric Digital Subscriber Line ("ADSL") connection or a wireless connection, for instance.

In that arrangement, the PSTN 22 may be connected to a Telephone Remote Access Concentrator ("TRAC") 24. In the data-over-cable system having an upstream telephone connection, the TRAC 24 may be a Total Control telephone hub by 3Com Corporation of Santa Clara, for instance. However, the TRAC 24 could also be a telephone hub manufactured by a different company, or could take still other forms.

The combination of the CMTS 12 and the TRAC 24 is called a "Telephone Return Termination System" ("TRTS") 26. The TRTS 26 is illustrated as a dashed box in Figure 1. The CMTS 12 and the TRAC 24 may be at a "headend" of the cable system 10. Alternatively, for instance, the TRAC 24 may be located in a different location and may have routing associations with the CMTS 12. The cable system 10 may also include a plurality of servers such as operations servers, administrative servers or maintenance servers (not shown). Further, the CMTS 12 may connect a plurality of access points to the data-over-cable system 10.

Additionally, the plurality of access points may be connected to cable headend access points. Such configurations may be "one-to-one", "one-to-many", or "many-to-many", and may be interconnected to other Local Area Networks ("LANs") or Wide Area Networks ("WANs").

The data-over-cable system 10 may comprise a plurality of network interfaces. As shown in Figure 1, the TRAC 24 is connected to a data network 28 (e.g. the Internet, an intranet, a LAN or a WAN, for instance) via a TRAC-Network System Interface 30 ("TRAC-NSI"). The CMTS 12 is connected to the data network 28 via a CMTS-Network System Interface ("CMTS-NSI") 32.

Further, the data-over-cable system 10 may comprise a policy/authorization server 38 in communication with the CMTS 12. The authorization/policy server 38 may manage overall policies with an administrative domain such as an Internet service provider, for instance. The CMTS 12 may also comprise an internal authorization module that may serve as a policy enforcement point, for instance.

The system 10 may also comprise a bandwidth manager 36 in communication with the CMTS 12. The bandwidth manager 36 may detect network trends, measure network response time, generate CoS and QoS reports, allocate bandwidth and/or keep records of allocated and available bandwidth.

The present invention is not limited to use within the data-over-cable system illustrated in Figure 1. More, fewer or different components, connections and interfaces could also be used.

Further, the arrangements described herein are shown for purposes of illustration only, and those skilled in the art will appreciate that other arrangements and other elements, such as interfaces or functions, whether or not known in the art, can be used instead, and some elements may be omitted altogether. Additionally, as in most communications applications, those skilled in the art will appreciate that many of the elements described herein are functional entities that may be

implemented as discrete components or in conjunction with other components, in any suitable combination and location.

Further, as mentioned above, network entities in the data-over-cable system 10 may be duplicated to provide a back-up in case of failure of one or more network entities. For instance, the network entities may be duplicated in parallel or in series. In a parallel arrangement, for instance, the CMTS 12 comprising an internal authorization/policy server 38 and an internal bandwidth manager 36 may be duplicated. The CMTS 12 and a duplicated CMTS 12' (not shown) may operate simultaneously, with one of them active and the other one in a "standby" state. In such an arrangement, the two units may communicate using a "keep alive" signal, for instance. Thus, if the primary CMTS 12 fails, the redundant CMTS 12' may immediately start operating, and, ideally, there is no loss of service.

In another exemplary embodiment providing a back-up system, redundant units may operate in a serial manner. In the serial arrangement, units may be cross-connected with a heart-beat controlled shunt on ports. Further, in the serial arrangement, both units may be active, as opposed to a primary device being in an active state and a redundant device being in a standby state, as in the parallel arrangement. In another exemplary embodiment of the present invention, any individual integral components or groups of components may be duplicated.

An operating environment for each CMTS 12, CM 16, CPE 18, TRAC 24 and other network entities of an exemplary embodiment may include a processing system with at least one high speed processing unit and a memory system. In accordance with the practices of persons skilled in the art of computer programming, the present invention is described below with reference to acts and symbolic representations of operations or instructions that are performed by the processing system, unless indicated otherwise. Such acts and operations or instructions could be referred to as being "computer-executed", "processing unit executed", or the like.

It will be appreciated that the acts and symbolically represented operations or instructions include the manipulation of electrical signals by the processing unit. An electrical system with data bits causes a resulting transformation or reduction of the electrical signal representation, and the maintenance of data bits at memory locations in the memory system to thereby reconfigure or otherwise alter the processing unit's operation, as well as other processing of signals. The memory locations may be physical locations that could have particular electrical, magnetic, optical, or organic properties for maintaining data bits.

The data bits may also be maintained on a computer readable medium such as magnetic disks, optical disks, organic disks, and any other volatile or non-volatile mass storage system readable by the processing unit, for instance. The computer readable medium may include cooperating or interconnected computer readable media, which may exist exclusively on the processing system or may be distributed among multiple interconnected processing systems that may be local or remote to the processing system.

Network device protocol stack

Figure 2 is a block diagram illustrating an exemplary protocol stack 36 for network devices in the data-over-cable system 10. In an exemplary embodiment of the present invention, network entities in the data-over-cable system 10 may be DOCSIS compliant. However, other standards may also be used, and the present invention is not limited to DOCSIS compliant network entities.

Figure 2 illustrates downstream and upstream protocols used in the CM 16, for instance. As known in the art, the Open System Interconnection ("OSI") model may be used to describe computer networks. The OSI model consists of seven layers including, from lowest to highest, a physical layer, a data-link layer, a network layer, a transport layer, a session layer, a presentation layer and an application layer. The physical layer transmits bits over a communication link. The

data-link layer transmits error free frames of data. The network layer transmits and routes data packets.

In bi-directional data-over cable systems, the CM 16 is connected to the cable network 14 in a physical layer 38 via a Radio Frequency ("RF") Interface 40. In an exemplary embodiment of the present invention, for a downstream data transmission, the RF Interface 40 may have an operation frequency range of 50 Mega-Hertz ("MHz") to 1 Giga-Hertz ("GHz") and a channel bandwidth of about 6 to 8 MHz. However, other operation frequencies may also be used, and the invention is not limited to these frequencies. For an upstream transmission the RF Interface 40 may have an operation frequency range of about 5 MHz to 50 MHz. Further, the RF Interface 40 may use a signal modulation method, such as Quadrature Amplitude Modulation ("QAM"). As known in the art, the QAM is used as means for encoding digital information over radio, wire, or fiber optic transmission links. The QAM is a combination of amplitude and phase modulation and is an extension of a multiphase phase-shift-keying. The QAM may have any number of discrete digital levels typically including 4, 16, 64 or 256 levels. In an exemplary embodiment, QAM-64 may be used in the RF Interface 40. However, other operating frequencies and modulation methods could also be used, such as a Quadrature Phase Shift Keying ("QPSK") modulation, for instance. Further, the RF Interface 40 can also be used in a cable system with a telephone return.

In a data-over-cable system with a telephone return employed for an upstream connection, the CM 16 may be connected to the PSTN 22 in the physical layer via a telephone interface 48. In an exemplary embodiment, the telephony interface may operate in accordance with one of the standards of the International Telecommunications Union-Telecommunication ("ITU-T") Standardization Sector. The telephone interface 48 may use the ITU-T V.90 standard, for instance. As known in the art, the ITU-T V.90 standard is commonly used in a data link layer

of modem communications, and it currently allows data rates as high as 55,600 bits-per-second ("bps"). However, the telephone interface 48 may also operate according to other communications standards, such as V.32 standard, V.34 standard or V.90 standard, for instance. Further, the telephone interface 48 could also be an Asymmetric Subscriber Link ("ADSL") interface, an Integrated Services Digital Network ("ISDN") interface or a wireless interface, for instance.

Above the RF Interface 40, there is a data link layer comprising a Medium Access Control ("MAC") layer 44. As known in the art, the MAC layer 44 controls access to a transmission medium via the physical layer 38. The MAC layer 44 may use a protocol described in IEEE 802.14. However, other MAC layer protocols could also be used, such as MCNS MAC layer protocol, for instance. Above the MAC layer 44 may be a link security protocol stack 46. The link security protocol stack 46 prevents unauthorized users from making a data connection from cable network 14.

A Point-to-Point Protocol ("PPP") layer 50 is in the data link layer 42 and above the telephone interface 48. As known in the art, the PPP layer encapsulates network layer datagrams over a serial communication link. More information on the PPP protocol may be found on the World Wide Web at the URL "www.ietf.org" in a Request for Comments ("RFC"), RFC-1661.

A network layer 52 is above both the downstream protocol layer and the upstream protocol layer. The network layer 52 comprises an Internet Protocol ("IP") layer 54 and an Internet Control Message Protocol ("ICMP") layer 56. The IP layer 54 corresponds to the OSI layer 3, which is the network layer, but, typically, is not defined as part of the OSI model. As known in the art, IP is a routing protocol designed to route traffic within a network or between networks. More information on the IP protocol may be found at the URL "www.ietf.org" in RFC-791. The ICMP layer 56 is used for network management. The ICMP provides a plurality

of functions, such as error reporting, reachability testing (e.g., "pinging"), congestion control, route-change notification and performance or subnet addressing, for instance. More information on ICMP may be found at the URL "www.ietf.org" in RFC-792.

A transport layer 58 is above the network layer 52. The transport layer 60 comprises a User Datagram Protocol ("UDP") layer 60, which approximately corresponds to the OSI layer 4, the transport layer. As known in the art, UDP provides a connectionless mode of communications with datagrams. More information on the UDP layer 60 may be found at the URL "www.ietf.org" in RFC-768. However, the transmission layer 58 is not limited to the User Datagram Protocol and other protocols, such as a Transmission Control Protocol ("TCP"), for instance. More information on the TCP may be found at the URL "www.ietf.org" in RFC-793.

Above the transport layer 58, there are a Simple Network Management Protocol ("SNMP") layer 60, a Trivial File Transfer Protocol ("TFTP") layer, a Dynamic Host Configuration Protocol ("DHCP") layer 66 and a UDP manager 68. The SNMP layer 60 is used to support network management functions. More information on the SNMP layer may be found at the URL "www.ietf.org" in RFC-1157. The TFTP layer 64 is a file transfer protocol, which is typically used to download files and configuration information. More information on the TFTP layer 64 may be found at the URL "www.ietf.org" in RFC-1350. The DHCP layer 66 is a protocol for passing configuration information to host on the IP network 54. More information on the DHCP layer 66 may be found at the URL "www.ietf.org" in RFC-1541, RFC-2131 and RFC-2132. The UDP manager 68 distinguishes and routes packets to an appropriate service. However, more, fewer, or different protocol layers could be used in the data-over-cable system 10.

According to an exemplary embodiment of the present invention, the CM 16 may support transmission and reception of IP datagrams as specified by RFC-791. The CM 16 may be also

configured to filter IP datagrams with IP addresses assigned to the CM 16 or CPE 18. Further, the CMTS 12 and the TRAC 24 may also perform filtering of IP datagrams.

During the initialization of the CM 16, the CMTS 12 transmits to the CM 16 a Termination System Information ("TSI") message, which is a MAC management message. The CMTS 12 may use the TSI message to report to the CM 16 whether or not a bi-directional system is used, for instance. Further, the TSI message may be used to provide the CM 16 with information about the status of the CMTS 12.

Figure 3 is a block diagram illustrating a structure of an exemplary TSI message 76. The TSI message structure 76 comprises a plurality of fields, such as a MAC management header 78, a downstream cable channel IP address 80, a registration IP address 82, a CMTS boot time 84, a downstream channel identifier 86, an epoch time 88 and vendor specific encoded data 90. However, the TSI message 76 is not limited to these fields, and more, fewer or different fields could also be used in the TSI message 76.

Additionally, during the initialization process, the CM 16 may initiate a Dynamic Host Configuration Protocol ("DHCP") process. The DHCP process is used to provide configuration parameters to hosts on a network such as an IP network, for instance. The DHCP process provides two main services to network clients such as CMs or CPE entities. First it allocates IP network addresses to clients and, second, the DHCP process provides configuration parameters for network entities.

Figure 4 is a block diagram illustrating an exemplary DHCP message structure 92. The format of the DHCP message structure 92 is based on a format of BOOTstrap Protocol ("BOOTP") messages described in RFC-951 and RFC-1542, incorporated herein by reference. From a network host client's point of view, the DHCP is an extension of a BOOTP mechanism.

This property allows the existing BOOTP clients to communicate with DHCP servers without requiring any changes in to network host clients' BOOTP initialization software.

To capture a BOOTP relay agent behavior described as part of the BOOTP specification and to allow interoperability of existing BOOTP clients with the DHCP servers, a DHCP server uses a BOOTP message format. Further, using the BOOTP relay agents eliminates the necessity of using a DHCP server on each physical network segment.

DHCP 66 message structure 92 includes an operation code field 94 ("op"), a hardware address type field 96 ("htype"), a hardware address length field 98 ("hlen"), a number of hops field 100 ("hops"), a transaction identifier field 102 ("xid"), a seconds elapsed time field 104 ("secs"), a flags field 106 ("flags"), a client IP address field 108 ("ciaddr"), a your IP address field 110 ("yiaddr"), a server IP address field 112 ("siaddr"), a gateway/relay agent IP address field 114 ("giaddr"), a client hardware address field 116 ("chaddr"), an optional server name field 118 ("sname"), a boot file name 120 ("file") and an optional parameters field 122 ("options"). Descriptions for an exemplary DHCP message 92 fields are shown in Table 1.

DHCP Parameter	Description
OP 94	Message op code / message type. 1=BOOTREQUEST, 2=BOOTREPLY.
HTYPE 96	Hardware address type (e.g., '1' = 10 Mps Ethernet).
HLEN 98	Hardware address length (e.g. '6' for 10 Mbps Ethernet).
HOPS 100	Client sets to zero, optionally used by relay-agents when booting via a relay-agent.
XID 102	Transaction ID, a random number chosen by the client, used by the client and server to associate messages and responses between a client and a server.
SECS 104	Filled in by client, seconds elapsed since client started trying to boot.
FLAGS 106	Flags including a BROADCAST bit.
CIADDR 108	Client IP address; filled in by client in DHCPREQUEST if verifying previously allocated configuration parameters.
YIADDR 110	'Your'(client) IP address.
SIADDR 112	IP 54 address of next server to use in bootstrap; returned in DHCP OFFER, DHCPACK and DHCPNAK by server.
GIADDR 114	Gateway relay agent IP 54 address, used in booting via a relay-agent.
CHADDR 116	Client hardware address (e.g., MAC layer 44 address).
SNAME 118	Optional server host name, null terminated string.
FILE 120	Boot file name, terminated by a null string.
OPTIONS 122	Optional parameters.

Table 1.

The DHCP message structure 92 shown in Figure 4 may be used to discover IP network host interfaces in data-over-cable system 10. A network host client such as the CM 16 may use the DHCP process to dynamically acquire or verify an IP address and network parameters whenever the network parameters may have changed. During a typical use DHCP process, the CM 16 and CPE 18 broadcast a "DHCPDISCOVER" message to receive configuration settings such a configuration file and an IP address for the CM 16 and an IP address for the CPE 18. During the DHCP process, the DHCP servers may respond with DHCPOFFER messages including configuration parameters. Then, the CM 16 or the CPE 18 selects one of the DHCP servers and sends a DHCPREQUEST to the selected server.

Policy management and multiple access provisioning

According to an exemplary embodiment, a system and method for the cable modem boot file management, IP service classes management, DHCP filtering, DHCP forwarding, CM setting recording and automatic provisioning in the existing data-over-cable system are developed.

Figure 5 is a block diagram illustrating an exemplary network system 142 for policy management and multiple access provisioning according to an exemplary embodiment. In one embodiment, the network system 142 is a data-over-cable system. However, other network systems could also be employed. The exemplary data-over-cable system 142 includes the CPE 18, the CM 16, the CMTS 12, a server cluster 147 and an administration tool 141 in communication with the server cluster 147. However, more fewer or equivalent components can also be used. The server cluster 147 includes a DHCP server 144 such as a Windows 2000 DHCP server, an Application Programming Interface ("API") Layer 148, a provisioning/access manager 146 and a database 150. According to an exemplary embodiment, the DHCP server 144 is configured with different scopes for CMs and different scopes for each CPE class of

service. Further, the database 150 stores DHCP configuration settings and configuration data for all registered CMs. According to an exemplary embodiment, the administration tool 141 communicates with the server cluster 147 using standard methods such as Open Data Base Connectivity ("ODBC") method for sharing data between databases and other programs. The administration tool 141 includes a Graphical User Interface ("GUI") tool 140, a Command Line Interface ("CLI") 143 and a Component Object Model ("COM") Administration Object API 145. According to an exemplary embodiment, the GUI 140 provides the ability to query, add, delete and modify configuration settings for individual CMs that are classified using a unique CM MAC address for each CM. The CLI 143 provides all GUI administration functional operations through a command line interface. The COM Administration Object API 145 provides the access to the administration functionality and allows for configuring CM boot files, service classes and correlating CM MACs to PC IP addresses.

Network devices for preferred embodiments of the present invention include network devices that can interact with network system 142 based on standards proposed by the Data- Over-Cable-Service-Interface- Specification ("DOCSIS") standards from the Multimedia Cable Network Systems ("MCNS"), the Institute of Electrical and Electronic Engineers ("IEEE"), International Telecommunications Union-Telecommunication Standardization Sector ("ITU"), Internet Engineering Task Force ("IETF"), and/or Wireless Application Protocol ("WAP") Forum. However, network devices based on other standards could also be used. DOCSIS standards can be found on the World Wide Web at the Universal Resource Locator ("URL") "www.cablemodem.com." IEEE standards can be found at the URL "www.ieee.org." The ITU, (formerly known as the CCITT) standards can be found at the URL "www.itu.ch." IETF standards can be found at the URL "www.ietf.org." The WAP standards can be found at the URL "www.wapforum.org."

According to an exemplary embodiment, a system administrator may manage the data-over-cable system 142 shown in Figure 5. However, prior to using this system, the administrator is required to define network interface configuration and DHCP options configuration. According to an exemplary embodiment, a set of commands such as a "set dhcp" command may be created on a system for system administrators to enter necessary DHCP options configuration parameters. In one embodiment, the DHCP options configuration parameters may include a two-way CM's gateway interface address ("cmgiaddr") and a CPE gateway interface address ("cpegiaddr"). The "cmgiaddr" corresponds to a name of a cable IP network through which the administrator wants DHCP responses to be routed for downstream transmission to the CM 16, and the "cpegiaddr" corresponds to a name of the cable IP network through which the administrator wants DHCP responses to be routed to CPEs such as the CPE 18. Further, according to an exemplary embodiment, system administrators have the ability to enable and disable policy management and multiple access provisioning system. In one embodiment, a system administrator may set a special QoS parameter such as an "agentinfooption" parameter in order to enable policy management and multiple access provisioning methods associated with the system.

Figure 6 is an exemplary dialog box that a system administrator uses to administer the provisioning/access manager 146. According to an exemplary embodiment, as described in greater detail in the preceding paragraphs, the system administrator manages: boot file mappings, service classes, DHCP filtering, DHCP forwarding and global service options. Using the interface, the system administrator can also select one of the graphical selection inputs to load cable modem configuration parameters from a file, to dump cable modem's configuration parameters to a file and to manage a license of the provisioning/access manager 146.

According to an exemplary embodiment, the system administrator may create and manage assignment of boot files to each CM based on MAC network addresses associated with the CMs, so that a CM can be uniquely identified and directed to appropriate configuration settings. Thus, based on a CM's customer service plan, a CM can be assigned a unique boot file that may add, delete, update or restrict network-based features according to the system administrator's settings. In one embodiment, the system administrator may specify a "BootFileID", a "BootFile Path" and may enter a brief description of the boot file via the graphical user interface 140. According to an exemplary embodiment, the "BootFileID" corresponds to a unique identifier of a CM's boot file and is added to the configuration settings when a new CM MAC address is added to the system. The "BootFile Path" is a full path to the boot file located on a TFTP server and, according to an exemplary embodiment this path is inserted into an outgoing DHCP packet for the CM. Table 2 shows an exemplary boot file configuration data. Figure 7 illustrates an exemplary dialog box that a system administrator uses to manage configuration file settings.

Boot File ID	TFTP Path	Description
1	black.cfg	Unknown CM default .cfg file
2	Standard.cfg	Standard CoS .cfg file
3	gold.cfg	Gold CoS .cfg file

Table 2.

Further, according to an exemplary embodiment, CPEs in the data-over-cable system 142 can be provisioned so that certain pools of IP addresses have priority over others and different CoS can be mapped to different IP address pools. As known in the art, the CMTS 12 is configured to have one gateway address "cpegiaddr" for all CPE requests, and such configuration works well if the CMTS 12 has only one downstream for CPE addresses. However, such configuration does not work if the CMTS 12 has multiple downstreams for CPE

addresses having different classes of service. Since each CMTS is configured to have one “cpegiaddr” for all CPE requests, an IP address corresponding to a desired class of service may be out of the requesting CPE’s subnet. According to an exemplary embodiment, a different gateway address (mapped “giaddr”) is placed in the incoming DHCP request associated with a CPE based on the class of service required by each CPE. The system administrator may create a set of rules for mapping an incoming “giaddr” (the “cpegiaddr” in the CMTS) to a mapped “giaddr”. Table 3 shows an exemplary service class mappings that may be created by the system administrator via the graphical user interface 140 and then stored in the database 150. Figure 8 is an exemplary dialog box that a system administrator uses to manage class of service settings.

Incoming giaddr	Service Class	Mapped giaddr	Subnet Mask	Description
149.112.10.1	11	149.112.10.1	255.255.255.0	Black service class, unknown CM
149.112.10.1	12	149.112.11.1	255.255.255.0	Standard service class
149.112.10.1	13	149.112.12.1	255.255.255.0	Gold service class

Table 3.

As shown in Table 3, the system administrator sets a plurality of fields while configuring service classes. One of the fields corresponds to an incoming “giaddr”, which in an exemplary embodiment corresponds to the “cpegiaddr” defined by the system administrator during setting the DHCP options configuration. Further, the system administrator sets service classes, which are defined by numeric identifiers such as 11, 12 and 13 shown in Table 2. The numeric identifiers do not have to be unique, and the unique key is a combination of an incoming “giaddr” and service class. The mapped “giaddr” field corresponds to a CPE’s downstream IP address that is mapped from the incoming “giaddr”. The subnet mask field corresponds to a subnet mask for a specific service class.

According to an exemplary embodiment, system administrators may also restrict vendors and versions of the CMs on their networks. In one embodiment, system administrators set MAC prefixes that are allowed on their network. For example, a system administrator may set three byte MAC prefix values. However, the exemplary embodiment is not limited to the three byte MAC prefix values, and other prefix values could also be used such as four byte prefix values, for example. If the filtering is enabled by a system administrator, the provisioning/access manager 146 scans all requests to determine whether or not the first three bytes of the CM MAC address reside in the table defining the allowed MAC prefixes. If the prefix is not found in the database, the packet is dropped and the CM does not receive network services such as an IP address assignment or configuration file assignment. Table 4 shows an exemplary set of MAC prefixes associated with the CMs' MAC addresses that could be set by the system administrator. Figure 9 illustrates an exemplary dialog box that the system administrator can use to manage and set MAC address prefixes using the graphical user interface tool 140.

MAC Prefix	Description
0x00FF11	Generic 1
0x801122	Company X

Table 4.

Further, according to an exemplary embodiment, a system administrator may set DHCP forwarding functions when configuring the server cluster 147. The DHCP forwarding function allows CPE DHCP requests or both CPE and CM DHCP requests to be forwarded to an external DHCP server. In one embodiment, the DHCP forwarding is based on the CM MAC address. A system administrator may assign to each CM MAC address a provider ID that maps to a DHCP server's IP address. Table 5 illustrates an exemplary set of service provider data that may be set, updated and deleted for a specific CM MAC address by the system administrator. Figure 10 is an exemplary dialog box that a system administrator uses to manage DHCP forwarding.

Provider ID	IP Address	Description
3	20.21.22.23	Provider X
4	29.21.22.23	Provider Y

Table 5.

If the DHCP forwarding function is enabled, the DHCP server 144 simply forwards the DHCP requests to the specified DHCP server and does not attempt further processing of the packet. According to one embodiment, the default condition is to use the DHCP server 144 that is co-resident with the provisioning/access manager 146. However, if forwarding is enabled, subscriber's requests still have the appropriate policies processed as set for the subscriber and, then, they are forwarded based on the attributes of individual CMs.

In order to set up a network, a system administrator may also configure service-wide options. In one embodiment, the system administrator may manage and set up default parameters for each non-registered CM. Further, the system administrator may set up global service option parameters. Table 6 shows an exemplary set of parameters with a description of each parameter that could be managed by the system administrator. In one embodiment, the system administrator may disable or enable service-wide option using a graphical user interface that displays to the system administrator the options as graphical selection inputs. Figure 11 is an exemplary dialog box that a system administrator uses to manage the service-wide options.

Parameter	Description
Boot File ID for 2 Way CM	Default boot file assigned to 2-way CMs that make DHCP requests but are not yet provisioned in the database 150
Service Class	Default Service Class assigned to CMs that make DHCP requests but are not already provisioned in the database 150
Boot File ID for Telco Return	Default bootfile ID assigned to telco-return CMs that make DHCP requests but are not already provisioned in the database 150
Provider ID	Default provider ID assigned to CMs that use other DHCP servers but are not already provisioned in the database 150
Enable CM Filtering	Enables or disables DHCP filtering
Disable Lease Renewals	This feature is necessary for automatic

	provisioning. When enabled, all DHCP rebind requests are denied which forces CPEs and CMs to start the process over with a DISCOVER message
Enable Service Classes	Enables or disables the Service Class feature
Truncate RRAS MAC	Truncates all incoming hardware addresses to 6 bytes, blocking Routing and Remote Access Service ("RRAS") DHCP requests
Assign default bootfile	Add unknown CM to the database with the default boot file, service class, and provider ID
Default DHCP processing	DHCP server performs stock services for the request and does not add it to the database 150
Drop packet	Drops the packet
Disable forwarding	Disables DHCP forwarding to other DHCP servers
Forward all requests	Enables forwarding for CPE and CM requests
Forward CPE requests only	Enables forwarding for CPE requests only
CM/CPE Combo MAC key	The 12-character hex number identifying the CM/CPE combination card CM. The system uses this value to treat CM/CPE combination cards as both a CM and a CPE. This could be left blank.

Table 6.

In the exemplary data-over-cable network 142, "black.cfg" (BootFile ID=1), as shown in Table 2, has been defined as the unknown CM Boot File, and the Black service class (Service Class=11), as shown in Table 3, has been defined as the Service Class for the unknown CMs.

Further, according to an exemplary embodiment, the system administrator may save the CMs' configuration settings in the database 150. According to an exemplary embodiment, the configuration settings are saved in the database 150 as text files, and a record is created for each registered CM in the data-over-cable network 142. For example, an exemplary format of a CM record is: "M,0xFFFFFFFFFFFF,2,12,0,0". In the exemplary record, the first field represents an action or control code, where "M" represents an "add/modify" control code and "D" could represent a delete control code. The second field is "0x" followed by 12 hex characters that identify the CM MAC address being added or modified. The third field represents the Boot File ID of the boot file associated with the MAC network address of the CM specified in the second

field of the record. In the exemplary embodiment, the record specifies a Boot File ID 2 that, as shown in Table 2, corresponds to a standard boot file. The fourth field represents the Service Class ID (the service class that the CPE associated with the CM in the record should receive). In the exemplary embodiment, the record specifies the service class 12 that, as shown in Table 3, corresponds to a standard service class. The fifth field represents the Provider ID to which DHCP requests should be forwarded, and the sixth field represents Flags.

According to an exemplary embodiment, the system administrator may allow external systems to interact with the DHCP server 144 by creating programs that give an access to the server cluster 147. For example, other servers could interact programmatically with the server cluster 147 by using the COM interface 145. Further, according to an exemplary embodiment, a wrapper such as a C wrapper could be built around the COM 145 to support other interfaces such as Java Interfaces, for instance. The GUI Tool 140 can be accessed by any COM-accessible language, such as C++, VBScript, JScript or Java, for instance. However, the present invention is not limited to these languages, and other currently existing or later developed languages could also be used.

Figure 12 is a flow chart illustrating an exemplary method 160 for provisioning and access managing of a network device.

Referring to Figure 12, at step 162, a first network device receives a first message from a second network device. According to an exemplary embodiment, the first message includes a request for network services. Further, the first message includes a plurality of fields where one of the fields defines an identifier of the second network device. At step 164, the first network device marks the first message with an identifier of a network access device associated with the second network device. In an exemplary embodiment, the first network device maps the identifier of the network access device in a field of the first message that is different that the field

including the identifier of the second network device. In one embodiment, the second network device comprises the network access device. In another embodiment, the second network device communicates with the network access device.

At step 166, a third network device intercept the first message. According to an exemplary embodiment, the third network device intercepts the first message prior to any first protocol network server receives the first message. At step 168, the third network device determines the identity of the second network device using the identifiers in the first message. In one embodiment, the second network device comprises the network access device, and in such an embodiment, the identifier of the second network device is the same as the identifier of the network access device inserted in the first message by the first network device. Then, the third network device concludes that the second network device is the network access device. Alternatively, if the second network device communicates with the network access device, the identifiers included in the first message differ, and the third network device concludes that the second network device communicates with the network access device.

At step 170, the third network device manages an assignment of the configuration parameters for the second network device based on the identity of the network access device. In one embodiment, the third network entity queries a database to retrieve a configuration record associated with the identifier of the network access device. According to an exemplary embodiment, the database includes a plurality of configuration records, and each record is developed based on the identifier of the network access device. In one embodiment, the configuration record includes the identifier of the network access device, a configuration file identifier with a path of a configuration file on a second protocol network server and a service provider identifier with an IP address of a first protocol server associated with the service

provider. Further, the configuration record includes a class of service parameter associated with any network devices in communication with the network access device.

In one embodiment, if the second network device comprises the network access device, the third network device inserts the path of the configuration file associated with the configuration file identifier so that the second network device receives the preferred configuration file. Alternatively, the third network device forwards the first message to a service provider indicated in the record. Further, if the second network device communicates with the network access device, the third network device uses the class of service parameter to redirect the incoming message and to assign an IP network address to the second network device from a pool of IP addresses associated with the class of service parameter.

In the exemplary embodiment, the first network device comprises a CMTS 12, the network access device is the CM 16 and the third network device is the provisioning/access manager 146. In the exemplary embodiment, the second network device comprises the CM 16 if the second network device is the network access device. Otherwise, the second network device comprises the CPE. Further, the identifier of the network access device comprises a MAC address of the network access device, and the first message comprises a DHCP message. In the exemplary embodiment, the first protocol server comprises a DHCP server, and the second protocol server comprises a TFTP server.

Figure 13A and 13B show a flow chart illustrating an exemplary method 150 for configuring a cable modem such as the CM 16 according to an exemplary embodiment.

Referring to Figure 13A, at step 182, the CM 16 sends a first message in the data-over-cable system 142. According to an exemplary embodiment, the first message is a DHCP message such as a DHCP discover message or a DHCP request message, and the first message has a message structure as shown in Figure 4.

At step 184, a CMTS such as the CMTS 12 determines whether a system administrator has enabled a parameter associated with the QoS policy provisioning method. According to an exemplary embodiment, the CMTS 12 determines whether the "agentinfooption" parameter has been set by the system administrator. If the system administrator has not enabled the "agentinfooptions" parameter, at step 186, the method 180 terminates, and the first message is processed according to a standard DHCP method. However, if the system administrator has enabled the "agentinfooptions" parameter, at step 188, the CMTS 12 modifies the first message. According to an exemplary embodiment, the CMTS 12 marks the first message with an identifier of a network access device. In the exemplary embodiment, the CM 16 is the network access device and, thus, the CMTS 12 marks the first message by placing a MAC network address of the CM 16 in the "options" field 120 of the first message and forwards the message.

At step 190, a third network entity such as the provisioning/access manager 146 intercepts the first message. According to an exemplary embodiment, the provisioning/access manager 146 intercepts the first message prior to any network server such as a DHCP server receives the first message. In one embodiment, the provisioning/access manager 146 has a plurality of dynamic link library ("dll") extensions that execute a set of instructions upon the receipt of every DHCP message. In such an embodiment, when the provisioning/access manager 146 receives the first message, the provisioning/access manager 146 calls one or more dll extensions.

Then, at step 192, the provisioning/access manager 146 determines the identity of a network device associated with the first message. In one embodiment, the provisioning/access manager 146 may determine the identity of the network device by comparing the MAC network address in the "options" field with a MAC network address in the "chaddr" field of the first message. If the MAC network address in the "options" field is the same as the MAC network

address in the "chaddr", the the provisioning/access manager 146 concludes that the query came from a cable modem. However, if the MAC network address in the "options" field differs from the network hardware address in the "chaddr" field then, the the provisioning/access manager 146 recognizes that the query came from a CPE. In the exemplary embodiment associated with the method 180, the MAC network address in the "options" field is the same as the MAC network address in the "chaddr" field. Thus, the provisioning/access manager 146 concludes that the first message is associated with the cable modem, and in the exemplary embodiment, the first message is associated with the CM 16. If the values in the fields would differ, the third network entity would conclude that the first network device was a CPE, and Figure 14A and 14B illustrate such an embodiment.

At step 194, the provisioning/access manager 146 queries a database such as the database 150 to determine if any pre-configured records exist in the database for the CM 16. According to an exemplary embodiment, the provisioning/access manager 146 queries the database 150 using the MAC network address retrieved from the "options" field in the first message. According to an exemplary embodiment, the provisioning/access manager 146 may specify the type of the first network device in the query. Thus, herein, the provisioning/access manager 146 specifies that the first network device is a cable modem.

If a system administrator has enabled the DHCP filtering options while setting up the configuration options parameters, at step 196 in Figure 13B, the provisioning/access manager 146 determines whether a prefix of the MAC network address of the CM 16 is one of the allowed MAC network address prefixes stored in the database 150. If the prefix of the MAC network address is not found in the database 150, at step 198, the provisioning/access manager 146 does not process the first message, and the method 180 terminates. If the prefix of the MAC network address is one of the allowed MAC address prefixes, at step 200, the provisioning/access

manager 146 determines whether the database 150 includes a configuration information record associated with the MAC network address of the CM 16. If the database 150 does not have a configuration information record for the CM 16, the provisioning/access manager 146 determines whether the system administrator set up any default settings for unknown network devices. If such settings has been set up, the MAC network address of the CM 16 is entered to the database. Further, the provisioning/access manager 146 determines a default service class and a default configuration file ID with a full path to that file on a network server such as a Trivial File Transfer Protocol server. Further, the provisioning/access manager 146 determines a Provider ID of a network server such as a DHCP server that has been assigned to handle the assignment of default settings such as the assignment of default IP addresses to unregistered network devices. In the exemplary embodiment, the DHCP 144 handles the assignment of the default configuration settings. At step 202, the default settings are returned to the CM 16.

Alternatively, a system administrator can enable one of the configuration options such as the "Default DHCP Processing" option or the "Drop packet" option for unregistered network devices. As described in Table 6, if the system administrator enables the "Default DHCP Processing" option, a network server such as the DHCP server 144 performs stock services for all received requests but does not add the MAC network address associated with the requests to the database 150. Further, if the system administrator enables the "Drop packet" option, the requests from the unregistered network devices are simply dropped and are not processed in the data-over-cable system 147.

If the provisioning/access manager 146 has determined at step 200 that the record for the CM 16 exists in the database 150, at step 204, the provisioning/access manager 146 retrieves the record from the database 150. In one embodiment, the provisioning/access manager 146 determines an identifier of a configuration file and an identifier of a service provider such as an

identifier of a DHCP server to which the first message should be forwarded. In one embodiment, as shown in Table 2, each configuration file's identifier maps to a configuration file path, and the provisioning/access manager 146 inserts that configuration file path to the first message.

At step 206, the provisioning/access manager 146 determines whether the "Service Forwarding" configuration options parameter has been enabled by the system administrator. If the "Service Forwarding" option has been enabled and the configuration record associated with the CM 16 includes an identifier of a service provider to which the first network message should be forwarded, at step 210, the provisioning/access manager 146 forwards the first message to the specified provider. Thus, according to an exemplary embodiment, the network server such as the DHCP server 144 does not attempt further processing of the request in the first message simply forwards the first message to the specified provider. As described in reference to Table 5, each identifier of a service provider maps to an IP address of the service provider. Thus, the provisioning/access manager 146 uses an IP address of the service provider from the record to forward the first message, and the method 180 terminates. However, if the "Service Provisioning" option is disabled, at step 208, the provisioning/access manager 146 uses the pre-provisioned configuration file identifier and configuration file path to process the first message.

The method 180 has been described in reference to network devices shown in Figure 5. However, it should be understood that the present invention is not limited to these network devices, and more, fewer and equivalent network devices could also be employed to carry out the described method. Further, unless specified to the contrary, the steps of the flow chart may be taken in sequence other than that described, and more or fewer steps could be used.

Figure 14A and 14B shows a flow chart illustrating an exemplary method 250 for configuring a first network device such as the CPE 18 in communication a network access device such as the CM 16.

Referring to Figure 14A, at step 252, the CPE 18 sends a first message in a data-over-cable system such as the data-over-cable system 142 shown in Figure 5. According to an exemplary embodiment, the first message is a DHCP message such as a DHCPDISCOVER message or a DHCPREQUEST message and, the structure of the first message is as shown in Figure 4. Further, according to an exemplary embodiment, the first message includes either a request to discover an IP address for the CPE 18 or an actual request for an IP address.

At step 254, a CMTS such as the CMTS 12 determines whether a system administrator has enabled a parameter associated with the QoS policy provisioning method. According to an exemplary embodiment, the CMTS 12 determines whether the “agentinfooption” has been set by the system administrator. If the system administrator has not enabled the “agentinfooption” parameter, at step 256, the method 250 terminates, and the request in the first message is processed according to a standard DHCP method. However, if the system administrator has enabled the “agentinfooption”, the first message is processed according to the exemplary embodiment.

In one embodiment, a vendor option field is added to the CMTS 12 such that when the “agentinfooption” is enabled, an identifier of a network access device such as a MAC network address of a network access device such as the MAC network address of the CM 16 is added to any DHCP request. Thus, at step 258, the second network device modifies the first message by marking the first message with the identifier of the CM 16. In a preferred embodiment, the CMTS 12 maps a MAC network address of the CM 16 in the “options” field 120 of the first message and forwards the message. Further, the CMTS 12 places in the first message an IP address of the relay agent (the “cpegiaddr” set in the CMTS).

At step 260, a third network entity such as the provisioning access manager 146 intercepts the first message. According to an exemplary embodiment, the provisioning access

manager 146 intercepts DHCP messages prior to any DHCP network server receives the messages. Thus, the provisioning access manager 146 intercepts the first message prior to a DHCP server such as the DHCP server 144 receives the first message. In one embodiment, the provisioning access manager 146 includes a plurality of dll extensions that execute a set of instructions upon intercepting of any DHCP message. In such an embodiment, when the provisioning access manager 146 receives the first message, the provisioning access manager 146 calls one or more dll extensions to execute a set of instructions such as instructions to query a database for a record associated with the identifier included in the first message.

At step 262, the provisioning access manager 146 determines the identity of a network device that sent the first message. In one embodiment, the provisioning access manager 146 determines the identity of the network device by comparing the MAC network address in the "options" field of the first message with a MAC network address in the "chaddr" field of the first message. In the exemplary embodiment, the MAC network address in the "options" field is different than the MAC network address in the "chaddr" and, thus, this implies that a CPE in communication with the CM 16 is behind the request.

At step 264, the provisioning access manager 146 queries the database 150 to retrieve a configuration record associated with the MAC network address in the "options" field of the first message. According to an exemplary embodiment, since the request has been identified as originating from the CPE, the implied function of the provisioning access manager 146 is to redirect the first message to a special IP range or scope based upon which network access device the CPE is behind. Further, if the CPE requests the network services for the first time, the provisioning access manager 146 enters the MAC network address from the "chaddr" field into the database 150 and marks it as belonging to the CPE. In one embodiment, this event may be a

trigger point for the provisioning/access manager 146 to query the CMTS 12 for a relationship of the CPE with a unique network access device such as the CM 16 in the exemplary embodiment.

If the system administrator has enabled the DHCP filtering options while setting up the configuration options, at step 266 in Figure 14B, the provisioning access manager 146 determines whether a prefix of the MAC network address of the network access device is one of the allowed MAC network address prefixes stored in the database. If the prefix of the MAC network address is not found in the database, the provisioning access manager 146 does not process the request in the first message, and the method 250 terminates.

If the prefix of the MAC network address is one of the allowed prefixes, the provisioning access manager 146 determines whether the database 150 includes a configuration information record associated with the MAC network address of the network access device. If the database 150 does not include any records associated with the network access device, the method 250 terminates. If the configuration information record for the MAC network address specified in the "options" field exists in the database, at step 270, the provisioning access manager 146 retrieves the configuration record from the database 150.

Based on the retrieved record, at step 272, the provisioning access manager 146 determines what class of service any network device behind the network access device should receive. After the class of service has been determined, the provisioning access manager 146 looks up the service class table, which is keyed by the incoming "giaddr" and service classes. The exemplary set of parameters associated with the service class configuration was shown in Table 4.

At step 274, after the provisioning access manager 146 finds an attribute associated with the class of service for the CPE, the provisioning access manager 146 maps the incoming "giaddr" to a new "giaddr" ("mapped giaddr") that specifies the scope of network addresses

providing a desired class of service. The "mapped giaddr" allows the network server such as the DHCP server to assign an IP address for the CPE out of the server's subnet, and a new subnet associated with the "mapped giaddr" provides the desired class of service. Further, the "mapped giaddr" does not necessarily differ from the incoming "giaddr" and, in some embodiments, the "mapped giaddr" is the same as incoming "giaddr". Thus, according to an exemplary embodiment, the subnets may be created with the context of service class or, alternatively, with the context of service provider or any other functionality or accounting based grouping.

At step 276, the provisioning access manager 146 determines whether the "Service Forwarding" configuration option parameter has been enabled by the system administrator. If the "Service Forwarding" configuration parameter has been enabled and, further, the configuration information record associated with the MAC network address of the network access device includes an identifier of the service provider to which the requests should be forwarded then, at step 280, the provisioning access manager 146 forwards the request to the specified service provider. In such an embodiment, the network server such as the DHCP server 144 does not attempt to further process the request in the first message sent from the first network device. However, if the "Service Provisioning" configuration option is disabled, at step 278, the network server such as the DHCP server 144 processes the request in the first message.

The method 250 has been described in reference to network devices shown in Figure 5. However, it should be understood that the present invention is not limited to these network devices, and more, fewer or different network devices could also be employed to carry out the described embodiment. Further, unless specified to the contrary, the steps of the flow chart may be taken in sequence other than that described, and more or fewer steps could be also be used. Further, the exemplary method is not limited to modifying "giaddr" field, and other fields could also be modified.

Further, according to an exemplary embodiment, once configuration parameters and IP addresses are assigned to a network device, a record is created for such network device, and the record is updated every time a lease of a new address is given out to the network device. For example, in an exemplary embodiment, the database 150 stores such records. Table 7 illustrates an exemplary set of parameters that is created for each network device and stored in the database 150.

Field	Type	Description
IP Address	Int	IP address assigned
CmMAC	Binary	MAC address of the CM to which this IP address has been assigned. If this IP address is assigned to a CPE, then this value is the MAC address of the CM associated with the CPE.
AssignedMac	Binary	MAC address of an adapter that this IP address was assigned. This value could be the same as the CmMAC value.
Type	Int	Identifies the device type that is associated with the IP address. Exemplary values are 1 for a CM and 2 for a CPE.
LeaseExp	Int	C time that the lease will expire.

Table 7.

Further, according to an exemplary embodiment, a simple automatic web registration and session based prepaid registration are supported in the data-over-cable system 147. Figure 15 shows an exemplary block diagram of a data-over-cable system 290 for the automatic web registration and session based prepaid registration. The exemplary system 290 is a simplified system and is intended to illustrate a plurality of exemplary network devices employed in the automatic registration and the prepaid registration process. The exemplary system 290 includes the CM 16, the CPE 18, the DHCP server 144, an application server 292, an intranet web server

296, Operations Support System with a Business Support System ("OSS/BSS") 294 and a Remote Authentication Dial-In User Service ("RADIUS") 298. However, the present invention is not limited to these network devices, and more, fewer or equivalent network devices could also be used. As is known in the art, the OSS supports the daily operation of the telecommunication infrastructure, order negotiation, order processing, testing and billing. The BSS is one of the types of the OSS and is typically used by network administrators to manage business operations such as billing, sales management, customer-service management and customer databases. Further, as known in the art, a RADIUS is an authentication/accounting client/server based software system that is used by Internet Service Providers ("ISPs") to verify passwords and/or usernames of their clients.

Figure 16A and 16B illustrate an exemplary method 300 for automated provisioning and configuring a first network device such as the CM 16 and a second network device such as the CPE 18 associated with the first network device.

Referring to Figure 13A, at step 302, the first network device boots for the very first time. According to an exemplary embodiment, a third network entity such as the provisioning-access manager 146 queries a database such as the database 150 to determine whether the first network device is one of the registered network devices. Thus, the third network entity determines whether any configuration records exist for a network hardware address of the first network device. However, since the first network device is unknown to the system, no records exist for the first network device. In one embodiment, the third network entity creates a default record for the MAC network address of the first network entity. Next, at step 304, the third network entity assigns a default configuration parameter set to the first network device, and a default configuration file is delivered to the first network device.

At step 306, the second network device associated with the first network device boots for the first time and requests network services. According to the exemplary embodiment, by this time, the first network device is already configured with the default configuration parameters and the default record exists in the database for the first network device. Thus, when the second network device associated with the first network device requests the configuration parameters, the third network entity retrieves the record of the first network device from the database and, at step 308, assigns an intranet IP network address to the second network device. Thus, the second network device receives a non-routable IP address and is given an intranet access.

Then, according to an exemplary embodiment, at step 310, a customer is automatically redirected to an Intranet web server such as the Intranet web server 296. At step 312, the customer is queried to enter a desired CoS, billing information and account information, and the customer signs-up for the service. For example, during the sign-up process, the customer is queried to enter a password or a userid that the customer wants to use. To verify whether the selected userid or the password is available, the Intranet web server 296 communicates with the OSS/BSS 294.

At step 314 in Figure 16B, the Intranet web server 296 communicates with an application server such as the application server 292. In one embodiment, the server 296 provides sign-up information data of the second network device to the application server 292. To create any records for the second network device, the application server 292 determines what network access device is associated with the first network device. To do that, the application server 292 queries a first protocol server such as the DHCP server 144 for a MAC network address of the network access device, which in the exemplary embodiment is the MAC network address of the CM 16. After the MAC network address is obtained, at step 316, a new configuration record is created for the network access device associated with the MAC network address, and the

configuration record is stored in the database 150. According to an exemplary embodiment, the new configuration record includes a plurality of parameters such as a configuration file name for the network access device with a path to the configuration file on a designated server, and a service class identifier for network devices attached to the network access device. Further, a record is created to identify what service class the customer should receive. It is possible that the customer signed up for a prepaid service plan, in which a MAC network address of a customer's network access device is marked as such. Further, if a customer signed up for a monthly service plan, any network devices attached to such customer's network access device will be given a new class of service.

At step 318, the application server 292 adds the user's information data to a pool of users' database on the OSS/BSS 294. Further, the OSS/BSS 294 adds the user's information data to a database of an authentication network device such as a database of the RADIUS 298. At step 320, the application server 292 directs the first network device to re-boot. In one embodiment, the application server 292 employs a Simple Network Management Protocol ("SNMP") to instruct the first network device to re-boot. When the first network device re-boots, at step 322, the first network device configures its internal parameters using a new configuration file. According to an exemplary embodiment, the first network device sends a DHCP message, and the first network device is assigned the new configuration file according to the process described in reference to Figure 13A and 13B. Upon a completion of the process, the first network device is configured with a plurality of configuration parameters corresponding to a class of service associated with the first network device.

At step 324, when a lease of the temporary intranet IP address of the second network device expires, the second network device is assigned a new IP address based on the class of service in the configuration records stored in the database 150. The process of assigning an IP

address to a network device such as the second network device has been described in reference to Figure 14A and 14B.

The method 300 has been described in reference to network devices shown in Figure 15. However, the exemplary method is not limited to these network devices and fewer, more or equivalent network devices could be employed to carry out the described embodiment. Further, the steps of the flow chart may be taken in sequence other than that described, and more or fewer steps could also be used.

There is a plurality of functions that may be used by external network devices to access the database 150 and to perform a number of functional operations. According to one embodiment, a “GetVersion” function, “SetCmInfo” function, “GetCmInfo” function, “DeleteCm” function, “DumpCmInfoToFile” function, “GetCmInfoFromFile” function, “GetIpForCmMac” function, “GetCmMacFromIp” function, “GetNbrOfLic” function and “GetLastError” function are created to allow external systems the interaction with the DHCP server 144. The “GetVersion” returns a version of a currently used API as a whole number. An exemplary function declaration of the “GetVersion” function is: GetVersion() as integer.

The “SetCmInfo” function is used to create a new entry in the database 150 if a CM MAC address does not exist in the database 150. An exemplary declaration of the “SetCMInfo” function is: SetCMInfo (CmMacAddr [in] as string, BootFileID as integer, PCServiceClass [in] as integer, ProviderID as integer, Flags as integer) as Boolean. An exemplary code sequence that may be used to create the “SetCmInfo” function is shown in Table 8. However, the exemplary embodiment is not limited to the shown code sequence, and other code sequences could also be used.

```

Dim ErrorCode As Variant, ErrorDesc As Variant
Set obj = CreateObject("DOCSIS.AdminMain")
Dim rc As Integer
rc = obj.SetCmInfo(edSetMac.Text, edSetBootID.Text,
5   edSetClass.Text,
    edProviderIDIn.Text, edFlagsIn.Text)
If rc = 0 Then
    obj.GetLastError ErrorCode, ErrorDesc
    MsgBox "Error Code: " & ErrorCode & " Error Desc: " &
10   ErrorDesc
Else
    MsgBox "Success"
End If
End Sub

```

Table 8.

The “GetCmInfo” function retrieves information about a given CM. An exemplary declaration of the “GetCmInfo” function is: GetCmInfo (CmMacAddr [in] as string, BootFileID [out] as variant, PCServiceClass [out] as variant, ProviderID[out] as variant, Flags[out] as variant) as Boolean. An exemplary code sequence that may be used to create the “GetCmInfo” is shown in Table 9. However, the exemplary embodiment is not limited to the shown code sequence, and other code sequences could also be used.

```

25   Private Sub bnGetCmInfo_Click()
    Dim obj As Object
    Set obj = CreateObject("DOCSIS.AdminMain")
    Dim rc As Integer
    Dim lBootID As Variant, lClass As Variant, ProviderID As
30   Variant, Flags As
    Variant
    Dim CmSettings As Variant
    rc = obj.GetCmInfo(edGetMac.Text, lBootID, lClass,
    ProviderID, Flags,
35   CmSettings)
    edGetBootID.Text = lBootID
    edGetClass.Text = lClass
    edProviderID.Text = ProviderID
    edFlags.Text = Flags
40   edSettings.Text = CmSettings
    MsgBox ("Returned " & rc)

```


End Sub

Table 9.

The “DeleteCm” function removes a record associated with the specified cable modem from the system. An exemplary function declaration is: DeleteCm (CmMacAddr [in] as string) as Boolean. An exemplary code sequence for creating the “DeleteCm” function is shown in Table 10. However, the exemplary embodiment is not limited to the shown code sequence, and other code sequences could also be used.

```
Private Sub bnDelCm_Click()  
Dim obj As Object  
Set obj = CreateObject("DOCSIS.AdminMain")  
Dim rc As Integer  
rc = obj.DeleteCm(edDelCmMac.Text)  
MsgBox ("Returned " & rc)2  
End Sub
```

Table 10.

The “DumpCmInfoToFile” function dumps the entire contents of the CmConfig table to a comma-delimited file. The FullFilePath can be either a local path such as “c:\temp\somefile.txt” or a UNC name such as “\\someserver\somefile.txt”. An exemplary function declaration of the “DumpCmInfoToFile” function is: DumpCmInfoToFile (FullFilePath [in] as string) as Boolean. Table 11 shows an exemplary code sequence for creating the “DumpCmInfoToFile” declaration. However, the exemplary embodiment is not limited to the shown code sequence, and other code sequences could also be used.

```
Private Sub bnDump_Click()  
Dim obj As Object  
Set obj = CreateObject("DOCSIS.AdminMain")  
Dim rc As Integer  
rc = obj.DumpCmInfoToFile(edDumpFilePath.Text)  
MsgBox ("Returned " & rc)
```

End Sub

Table 11.

The "GetCmInfoFromFile" function retrieves CM configuration information from the specified file. The text file could have the same format as the text file in the
5 "DumpCmInfoToFile". If the MAC address already exists, the other fields will be set to the values in the file. An exemplary function declaration of the "GetCmInfoFromFile" is: "Bool
GetCmInfoFromFile (FullFilePath [in] as string) as Boolean. Table 12 shows an exemplary code sequence for creating the "GetCmInfoFromFile" declaration. However, the exemplary embodiment is not limited to the shown code sequence, and other code sequences could also be
10 used.

```
Private Sub bnGetFileInfo_Click()  
Dim obj As Object  
Set obj = CreateObject("DOCSIS.AdminMain")  
15 Dim rc As Integer  
rc = obj.GetCmInfoFromFile(edGetFilePath.Text)  
MsgBox ("Returned " & rc)  
End Sub
```

Table 12.

The "GetIpForCmMac" function gets an IP address for the specified MAC address. An exemplary function declaration for the "GetIpForCmMac" function is: Bool GetIpForCmMac
(CmMac as string, IpAddress[out] as variant). Table 13 shows an exemplary code sequence for
20 creating the "GetIpForCmMac" function. However, the exemplary embodiment is not limited to the shown code sequence, and other code sequences could also be used.

```
Private Sub bnIpmapping_Click()  
Dim obj As Object  
Set obj = CreateObject("DOCSIS.AdminMain")  
30 Dim rc As Integer
```

```

If Option1.Value = True Then 'GetCmMacFromIp
rc = obj.GetCmMacFromIp(edIpAddr.Text, vntTmp)
edCmMacIp.Text = vntTmp
End If
5 If Option2.Value = True Then 'GetAssignedMacFromIp
rc = obj.GetAssignedMacFromIp(edIpAddr.Text, vntTmp)
edCmMacIp.Text = vntTmp
End If
10 If Option3.Value = True Then 'GetIpForCmMac
rc = obj.GetIpForCmMac(edCmMacIp.Text, vntTmp)
edIpAddr.Text = vntTmp
End If
MsgBox ("Returned " & rc)
End Sub

```

Table 13.

The “GetCmMacFromIp” function gets the CM MAC address for the specified IP address. According to an exemplary embodiment, the IP address could be a CM’s IP address or an IP address of a PC attached to the CM. An exemplary function declaration for the “GetCmMacFromIp” function is: Bool GetCmMacFromIp (IpAddress as string, CmMAC[out] as string). Table 14 shows an exemplary code sequence for creating the “GetCmMacFromIp” function. However, the exemplary embodiment is not limited to the shown code sequence, and other code sequences could also be used.

```

25 Private Sub bnIpmapping_Click()
Dim obj As Object
Set obj = CreateObject("DOCSIS.AdminMain")
Dim rc As Integer
30 If Option1.Value = True Then 'GetCmMacFromIp
rc = obj.GetCmMacFromIp(edIpAddr.Text, vntTmp)
edCmMacIp.Text = vntTmp
End If
If Option2.Value = True Then 'GetAssignedMacFromIp
35 rc = obj.GetAssignedMacFromIp(edIpAddr.Text, vntTmp)
edCmMacIp.Text = vntTmp
End If
If Option3.Value = True Then 'GetIpForCmMac
rc = obj.GetIpForCmMac(edCmMacIp.Text, vntTmp)

```

```

edIpAddr.Text = vntTmp
End If
MsgBox ("Returned " & rc)
End Sub

```

5

Table 14.

The “GetAssignedMacFromIp” function gets a MAC address of a CM or a MAC address of a CPE for an IP address passed in. An exemplary function declaration for the “GetAssignedMacFromIp” function is: Bool GetAssignedMacFromIp(IpAddress as string, MAC[out] as variant). Table 15 shows an exemplary code sequence for creating the “GetAssignedMacFromIp” function. However, the exemplary embodiment is not limited to the shown code sequence, and other code sequences could also be used.

```

Private Sub bnIpmapping_Click()
Dim obj As Object
Set obj = CreateObject("DOCSIS.AdminMain")
Dim rc As Integer
If Option1.Value = True Then 'GetCmMacFromIp
rc = obj.GetCmMacFromIp(edIpAddr.Text, vntTmp)
edCmMacIp.Text = vntTmp
End If
If Option2.Value = True Then 'GetAssignedMacFromIp
rc = obj.GetAssignedMacFromIp(edIpAddr.Text, vntTmp)
edCmMacIp.Text = vntTmp
End If
If Option3.Value = True Then 'GetIpForCmMac
rc = obj.GetIpForCmMac(edCmMacIp.Text, vntTmp)
edIpAddr.Text = vntTmp
End If
MsgBox ("Returned " & rc)
End Sub

```

20

25

30

Table 15.

The “GetNbrOfLic” function gets a maximum number of licenses that may be configured in the data-over-cable system 142. An exemplary function declaration of the “GetNbrOfLic” function is: Bool GetNbrOfLic(MaxLics[out] as variant). Table 16 shows an exemplary code

35

sequence for creating the declaration of the "GetNbrOfLic" function. However, the exemplary embodiment is not limited to the shown code sequence, and other code sequences could also be used.

```
Private Sub bnGetLic_Click()  
Dim obj As Object  
Dim rc As Integer  
Dim Lics  
Set obj = CreateObject("DOCSIS.AdminMain")  
rc = obj.GetNbrOfLic(Lics)  
If rc < 0 Then  
MsgBox "You have a " & Lics & " CM entry license"  
Else  
MsgBox "An error has occurred."  
End If  
End Sub
```

Table 16.

The "GetLastError" function gets the error code and description for the last error to occur for an object instance. An exemplary function declaration for the "GetLastError" function is: Bool GetLastError(ErrorCode[out] variant, ErrorDesc[out] variant), where the "ErrorCode" identifies a number that uniquely identifies the error that occurred and the "ErrorDesc" includes a string description of the error that has occurred. Table 17 shows an exemplary code sequence for creating the "GetLastError" function. However, the exemplary embodiment is not limited to the shown code sequence, and other code sequences could also be used.

```
Private Sub bnGetLastError_Click()  
Dim obj As Object  
Dim rc As Integer  
Dim ErrorCode As Variant  
Dim ErrorDesc As Variant  
Set obj = CreateObject("DOCSIS.AdminMain")  
rc = obj.GetLastError(ErrorCode, ErrorDesc)  
If rc < 0 Then
```

```
MsgBox "Error Code: " & ErrorCode & " Error Desc: " &
ErrorDesc
Else
MsgBox "Get Last Error Failed."
End If
End Sub
```

Table 17.